PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06T 9/00

A1

(11) International Publication Number: WO 96/39681

(43) International Publication Date: 12 December 1996 (12.12.96)

(21) International Application Number: PCT/US96/08171

(22) International Filing Date: 31 May 1996 (31.05.96)

(30) Priority Data:
08/466,969 6 June 1995 (06.06.95) US

(71) Applicant: APPLE COMPUTER, INC. [US/US]; 1 Infinite Loop - MS: 38-PAT, Cupertino, CA 95014 (US).

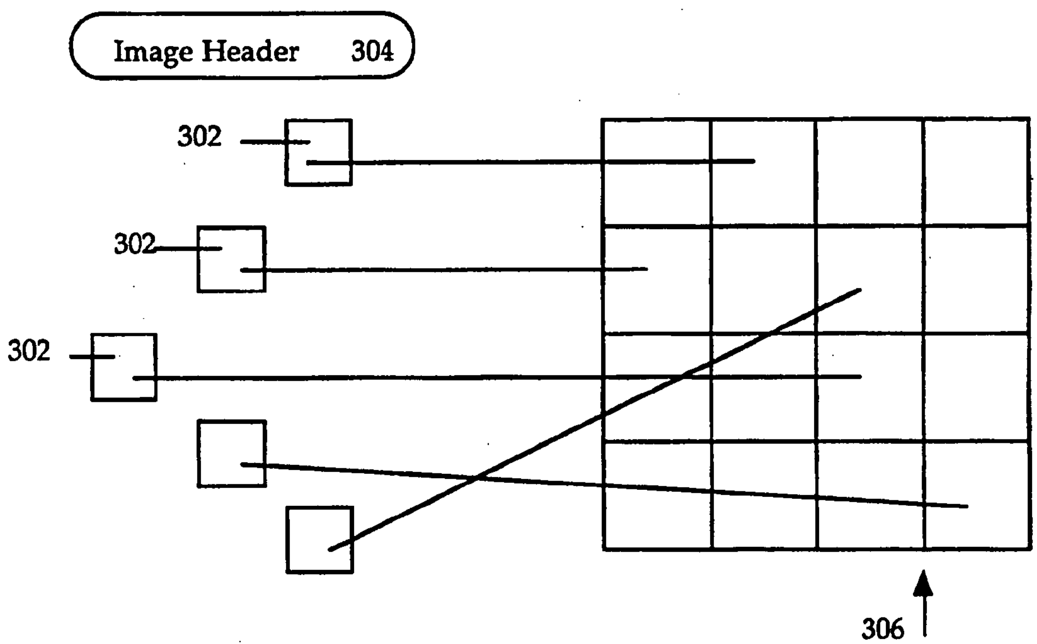
(72) Inventors: CHEN, Marilyn; 47 Walnut Avenue, Atherton, CA 94027 (US). CHAN, Allen, M.; 880 E. Fremont Avenue #628, Sunnyvale, CA 94087 (US). CHEN, Kok, S.; 870 E. El Camino Real #425, Sunnyvale, CA 94087 (US).

(74) Agents: CARMICHAEL, Paul, D. et al.; Apple Computer, Inc., 1 Infinite Loop - MS: 38-PAT, Cupertino, CA 95014 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: METHOD AND APPARATUS FOR PARTITIONING AN IMAGE



(57) Abstract

In accordance with exemplary embodiments, image data associated with a partitioned frame of input image data is stored along with information which identifies relative placement of each partitioned subimage within the frame of image data. The information is retained during processing of the input frame of image data, such that the original image can be reconstructed therefrom without the inclusion of holes or artifacts due to misalignment of boundaries. Thus, an input frame of image data which has been partitioned can be efficient transmitted, stored and rendered without concern that quality of the reconstructed image will be sacrificed.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgystan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

METHOD AND APPARATUS FOR PARTITIONING AN IMAGE

Technical Field

The present invention relates generally to processing of image data. More particularly, the invention relates to processing of image data by partitioning the image into a plurality of subimages.

Background

Digitized images displayed on scanners, monitors and printers are typically quantized on a pixel-by-pixel basis and stored in memory as pixel maps, more commonly referred to as pixmaps. A pixmap is a two-dimensional array of picture elements mapped to locations of the digitized image.

To provide sufficient color range and fidelity, each pixel is typically stored in memory as a set of color coordinates which define the pixel's color location in a given color space. For example, a pixel is typically represented by its red, green and blue color component values, or by its cyan, magenta and yellow color component values. The fidelity of the color reproduction is a function of the accuracy with which each color component is represented.

Due to the increased volume of data required to represent color images, effort has been directed to decreasing the memory requirements associated with color data pixmaps to provide more efficient transmission and storage of the image. A known technique for enhancing processing speed is to divide the input frame of image data into a plurality of subdivided, or partitioned areas, so that each subdivided area can be individually processed. However, a common problem encountered when an input frame of image data is divided into blocks is that pixels of the subdivided images are not properly transformed to an output space wherein the original image is to be reconstructed.

For example, Figure 3a illustrates an image represented as a frame of image data. This frame of image data is shown divided into a plurality of subdivided images which can be individually processed. By dividing the frame of image data to a plurality of subdivided images, memory requirements associated with processing components can be reduced since only a single subdivided image need be stored in memory at any given time during processing.

However, it is often desirable to transform the image represented by the input frame of image data in Figure 3a to a different scale or resolution for output (e.g., output on a display or printer). Further, it is often desirable that the image represented by the frame of input image data in Figure 3a be skewed

or rotated for subsequent output. For example, it may be desirable to scale a 300 pixel by 300 pixel input frame of image data to a printer output frame of 1000 pixels by 1000 pixels. Because 1000 is not an integer multiple of 300, a direct mapping of each pixel in the input frame to a set number of pixels in the output frame is not possible. As a result, some pixels in the output frame may be left unmapped thereby creating an appearance of holes in an output image which is produced using the output image data.

In addition, it may be desired to skew or rotate the input frame of image data to produce a rotated image at the output device. Those skilled in the art will appreciate that when the plurality of subimages illustrated in Figure 3a are skewed or rotated, some additional pixel locations in the output frame of image data will not be mapped to any subimage of the input frame of image data. That is, a rectangular subimage which encompasses a 3x3 pixel array of the input frame, when skewed or rotated by a fixed angle, will no longer be defined by a rectangular pixel array in the output frame. See, for example, Figure 2a wherein an image is represented by a plurality of pixels which form a square. However, once the image is skewed, it is no longer represented by a plurality of pixels which represent a square in the output frame buffer. Figure 2b shows a similar example for plural subimages which are skewed. Again, pixels which are not mapped to any of the subimages in the input frame of image data will appear as holes in the output image when it is displayed or printed.

Thus, computational errors, such as errors in rounding or truncation when an image is transformed, can result in an incomplete allocation, or mapping of pixels in an output frame of image data among the various subimages of the input frame of image data. If the pixel locations in an output frame buffer which stores the output image data are not accurately mapped to the various subimages of the input frame of image data, image artifacts will appear along boundaries between subimages when the original image is reconstructed, and will be readily apparent to the unaided eye.

A known technique for addressing the inaccuracies associated with processing partitioned image data is to divide the input frame of image data into several overlapping subimages so that gaps between subimages will not occur in the output frame buffer. However, such techniques do not resolve misalignments of boundary artifacts. For example, misalignments between the plurality of subimages in the output frame of image data can produce image artifacts due to the inaccurate representation of the boundaries between subimages.

Another technique for addressing the inaccuracies of transforming partitioned image data into an output frame buffer includes mapping the pixel locations between the input frame of image data and the output frame of image data to avoid boundary misalignments. Mapping techniques between an input space associated with the input frame of image data and an output space associated with the output frame of image data are of two general types: (1) those that perform image transformation from an input device space to output device space using forward mapping; and (2) those that perform image transformation from an input device space to an output device space using inverse mapping. These latter two techniques are described in a book entitled "Digital Image Warping", by George Wolberg, ISB0-8186-8944-7, 1984, pages 42-45.

Forward mapping generally refers to a mapping from an input image space to an output device space by copying each input pixel onto the output image in a frame of output image data at positions determined by X and Y mapping functions, as described for example in the aforementioned Wolberg book. Although forward mapping can efficiently map an input frame of image data to an output frame of image data, this technique suffers the drawback of having holes in the output image when reconstructed. For example, when an input image is resampled to produce an output image (e.g., scaled and/or rotated), forward mapping cannot accurately map every pixel in the output frame of image data to a particular pixel of a subimage in the input frame of image data. Rather, some pixels in the output image are left unmapped to a subimage of the input frame due, for example, to the rounding or truncation which occurs when the input image is transformed to the output space.

The inverse mapping technique is performed by computing a mapping of every output pixel to the input image. Although inverse mapping guarantees that no holes will result in the reconstructed image, this technique suffers the drawback of much slower performance because a computation must be performed for each pixel in the output frame of image data. That is, a computation is used to identify all pixels in the input frame of image data which are to be mapped to that particular output pixel.

Summary

To address the foregoing limitations associated with prior art systems, the present invention provides methods in accordance with independent claims 1 and 15, an apparatus in accordance with independent claim 12, and a computer readable medium in accordance with independent claim 14. Further

advantageous features, aspects and details of the invention are set forth in the dependent claims, the following description and the drawings. The claims are to be understood as a first non-limiting approach of defining the invention in general terms.

In accordance with exemplary embodiments, image data associated with a partitioned frame of input image data is stored along with information which identifies relative placement of each partitioned subimage within the frame of image data. The information is retained during processing of the input frame of image data, such that the original image can be reconstructed therefrom without the inclusion of holes or artifacts due to misalignment of boundaries. Thus, an input frame of image data which has been partitioned can be efficiently transmitted, stored and rendered without concern that quality of the reconstructed image will be sacrificed.

If an image contains significant amounts of data, such as with a large image, a high resolution image or an image where continuous tone (contone) color is desired, memory requirements associated with processing the image data can be significantly reduced, since each subimage can be processed individually. Subsequent processing (for example, compression of the output frame of data) can also be efficiently implemented. By initiating processing of subimages without having to wait for an entire frame of input image data, the rendering of the image can be performed during throughput of image data in a pipelined architecture. Further, those skilled in the art will appreciate that multiple processors can be exploited to operate upon different subimages of a partitioned image, and to thereby further enhance overall processing efficiency of the input image data without affecting quality of the reconstructed image. Exemplary embodiments achieve such advantages using a technique which provides performance equal to or better than that associated with conventional partitioning techniques, such as forward mapping.

Brief Description of the Drawings

Other objects and advantages of the present invention will become more apparent from the following detailed description of preferred embodiments when read in conjunction with the accompanying drawings wherein like elements are designated by like numerals and wherein:

Figure 1 illustrates a printing control system which incorporates an exemplary embodiment of the present invention;

Figures 2a and 2b illustrate exemplary image pixels in both input and output spaces;

Figures 3a-3c illustrate an image partitioned model and reconstructed image model; and

Figure 4 illustrates a flow chart of a method for processing image data in accordance with an exemplary embodiment of the present invention.

Detailed Description

Figure 1 illustrates an exemplary embodiment of an apparatus for processing an input frame of image data in accordance with an exemplary embodiment of the present invention. As referenced herein, image data corresponds to data obtained from an image that can be digitized for storage in a pixmap and subsequently compressed. Image data that is processed in accordance with exemplary embodiments of the present invention can be digitized image data captured through any instrumentation, such as a video camera.

Image data as referenced herein encompasses scanned data as well as non-scanned data. Non-scanned data typically includes character glyphs and graphical data, such as hand-sketched or computer generated graphics. Those skilled in the art will appreciate that image data processed in accordance with exemplary embodiments of the present invention can also include any combination of scanned and non-scanned data.

As referenced herein, a frame of image data corresponds to a preselected group of digitized data, such as the data associated with a display screen or a printed sheet. Typically, such image data is obtained from or reproduced using a scanner device, a monitor or a printing device. However, those skilled in the art will appreciate that the image data can merely be processed for transfer between two locations (for example, between two computers).

It will be appreciated that exemplary embodiments of the present invention can be applied to the processing of image data for reproduction using any visual medium. In the case of printing, the image will typically be reproduced on conventional size paper such as letter size, A4, B5 and legal size. However, those skilled in the art will appreciate that the present invention can be applied to images of any size which are reproduced in any format.

To facilitate an understanding of the present invention, Figure 1 illustrates a printing system 100 which incorporates features of the present invention. The Figure 1 printing system includes a color laser engine 102, such as any commercially available color laser marking engine. For purposes of the following discussion, the term "color" includes use of multiple colors (such as red, green and blue), as well as gray-scale printing using varying shades of gray.

Referring to Figure 1, an incoming frame of data is directed by a processor, represented as printing system controller 106 having associated memory 104, to a compression/decompression coprocessor 108 (CDC). Because the memory required to store an entire frame of image data within the printing system 100 is impractical, an entire frame of uncompressed data is not stored within the printing system 100 at any given time. Rather, a frame buffer stores the incoming frame of image data in portions (for example, on a row-by-row basis), for sequential processing. As each portion of the frame is processed, another portion is fetched by the printing system controller. As a result, the entire frame of image data is only stored in the print system 100 in compressed form.

In accordance with exemplary embodiments, the printing system controller 106 can be a reduced instruction set computer (RISC) such as the 33 Megahertz 29030 processor available from Advanced Micro Devices. The printing system controller 106 pre-processes an incoming frame of image data to: (a) scale the image data to a desired resolution; (b) partition the image data into partitioned blocks of a predetermined size; (c) resample the image data to ensure its alignment with the partitions; (d) filter the image data to ensure that each partitioned block contains no more than a predetermined number of color variations; and (e) create a side information buffer with information acquired during the pre-processing that can be used to enhance performance during subsequent processing.

In accordance with exemplary embodiments, the compressing of image data includes a step of classifying each partitioned block based on whether the partitioned block is determined to be color variant or determined to be color invariant. This step of classifying can be implemented by comparing the pixels within a given partitioned block to one another during the pre-processing to determine whether a given partitioned block is color variant or color invariant. Once having classified each partitioned block as being either color variant or color invariant, the compression/decompression co-processor 108 can be used to further compress the data. Those skilled in the art will appreciate that the steps of classifying each partitioned block as being either color variant or color invariant can be performed by the printing system controller 106 as part of the pre-processing (for example, classification information can be stored in the side information buffer), or can be performed by the compression/decompression coprocessor 108.

The compression/decompression coprocessor 108 can, for example, be formed as a monolithic application specific integrated circuit (that is, an ASIC

chip). However, those skilled in the art will appreciate that the processing implemented by the coprocessor 108 can be performed by the same processor used to implement the functions of the controller 106. The compression/decompression coprocessor compresses the image data included in the partitioned blocks of pixels to substantially reduce the memory requirements required to store a frame of image data.

In accordance with exemplary embodiments, the compression/decompression coprocessor compresses the image data within each partitioned block by representing a block which is determined to include color variations with less color fidelity than a block which is determined not to include color variations. Further, the compression/decompression coprocessor uses memory (that is, any specified memory) for storing the compressed image data as a representation of the original image. Alternately, the compressed image data can be immediately transmitted from the printing system 100 as it is compressed, for external buffering and decompression.

In an exemplary embodiment, a decompression engine is included within the compression/decompression coprocessor for use during the compression process. Recall that an entire frame of uncompressed image data is not stored in the printing system 100 at any given time; rather the entire frame is stored and compressed in sequentially processed portions. The decompression engine is provided within the compression/decompression coprocessor to accommodate a situation where newly received image data is to be superimposed on portions of the frame which have been earlier compressed. In this situation, the earlier compressed portion of the frame is retrieved, decompressed and returned to the frame buffer. The decompressed data which has been returned to the frame buffer is then overlaid with the newly received image data, after which the superimposed image data is recompressed.

Those skilled in the art will appreciate that the compression/decompression coprocessor need not be implemented using a separate chip; rather the compression/decompression functions can be implemented with any or all other functions of the Figure 1 system in a single ASIC using a single processor. Further, those skilled in the art will appreciate that the compression and decompression functions of the Figure 1 system can be implemented in software or hardware. In the exemplary Figure 1 embodiment, the compression functions are implemented using software and hardware, while the decompression functions are primarily implemented in hardware.

Once a frame of image data has been compressed and stored in the compression/decompression coprocessor, it can subsequently be transferred to the printer engine 102 via a system bus 110 and a video interface device (VID) 112. The video interface device can, for example, be formed as a separate ASIC chip having a decompression processor to support decompression and half-toning. Alternately, a single processor can be used to implement the functions of the controller 106, the coprocessor 108 and video interface device 112. The video interface device provides high quality reproduction of the original image from its compressed format.

The Figure 1 system further includes an input/output (I/O) communications device 114. The input/output communications device can include, for example, built-in networking support as well as parallel/serial I/O ports. Further, the I/O communications device can include additional memory as well as memory expansion ports. Any conventional I/O communications features can be used in accordance with the present invention, such that the I/O communications device need not be described in detail.

Uncompressed Image Data Format

An input frame of image data associated with an original image has a given width and length. The number of pixels in a given row of a scan line across the width of the frame is set in a horizontal total pixel count register. In accordance with exemplary embodiments of the present invention, the value set in the horizontal total pixel count register is divisible by 4.

In a vertical direction, along the length of the frame (and in the case of printing, along the paper motion direction), the number of pixels is set in a vertical total pixel count register. Again, the value set in the vertical total pixel count register of exemplary embodiments is divisible by 4.

In accordance with exemplary embodiments, the user can also set the number of bits used to represent each color component for a given pixel to 1, 4 or 8 bits. For example, if the user selects 4 color components to define a pixel, with each color component being represented by 8 bits, then each pixel would be represented as a 32-bit word (that is, 8 bits for each of the cyan, magenta, yellow and black color components).

In accordance with exemplary embodiments, the color of a pixel can be represented by any number of color components, including 1, 3 or 4 color components. For example, a four color component representation includes cyan, magenta, yellow and black color components. For a given application,

when the number of color components used to represent a given color is set to 1, the color of a pixel is defined by the magnitude of a single color component (such as black). When the number of color components is set to three, the color components used can be, for example, cyan, magenta and yellow color components. When the number of color components is set to be four, the 4 color components mentioned above can be used to define the color of a given pixel.

Where each of four color components in each pixel of a pixmap is encoded with 8-bits, a letter-size page having approximately 32 million, 600 dots per inch (dpi) color pixels, requires approximately 128 Megabytes of memory to store the page. Because a memory requirement of 128 Megabytes per page is cost prohibitive, exemplary embodiments of the present invention are directed to compressing this data using a partitioned input frame of image data.

Pre-processing of Image Data

As mentioned previously, image data which is received for storage in an image data input frame buffer of memory 104 is preprocessed. The preprocessing can include partitioning the frame into blocks having a predetermined number of pixels. Once partitioned further processing of the frame of image data can include: (a) scaling a frame of digitized image data to ensure that each partitioned block will have the same number of pixels; (b) resampling the image data to account for misalignment between the frame of image data and partitioned blocks and/or (c) any other desired processing of the data.

In accordance with the exemplary embodiment described herein, an input frame 300 of image data is partitioned, in a first level of partitioning, into blocks which represent subimages 302, as illustrated in Figure 3a. These subimages can, if desired, be further subdivided, in a second level of subpartitioning, to form partitioned areas formed as 4-by-4 pixel arrays within each subimage for purposes of enhancing subsequent processing. For example, the partitioned areas of each subimage can be compressed in a manner as described in commonly assigned U.S. Application Serial No. 08/397,372, filed March 2, 1995 (Attorney Docket No. P1481/149), entitled "METHOD AND APPARATUS FOR COMPRESSION OF DIGITIZED IMAGE DATA USING VARIABLE COLOR FIDELITY." The exact block size can be set by the user as a block size field stored in a register.

Those skilled in the art will appreciate that both the partitioned subimages and the partitioned blocks can have any number of pixels, and that the subimages and blocks can be of any desired shape. Alternately, the partitioned subimages can be considered the subpartitioned blocks if no further subdividing is desired. Accordingly, for purposes of simplifying the following discussion, only the partitioning of an image into subimages will be discussed with respect to the subdivided images. Those skilled in the art will appreciate that features described with respect to the subdivided images can be applicable to any one or more levels of partitioning.

In an exemplary embodiment, the subimages are capable of being tiled in non-overlapping fashion to encompass the entire image. For example, the shape of each partitioned subimage need not be square, but can be rectangular, cross-shaped, or shaped as described in "Tilings and Patterns, an Introduction", by W. H. Freeman, New York, 1989, ISBN 0-7167-1998-3. For example, the skewed patterns of Figures 2a and 2b can be specified as a tile shape for the subimages in a frame of image data.

Those skilled in the art will appreciate that such partitioning is not limited to a single shape, but can be a combination of shapes. Further, those skilled in the art will appreciate that non-tiled and/or non-overlapped blocks can be used at some possible expense to the compressed buffer size and/or the integrity of the compressed data relative to the original image.

In accordance with exemplary embodiments, enhanced processing of partitioned image data is performed in a manner which permits efficient processing without sacrificing the quality of a reconstructed image. In accordance with exemplary embodiments, the processor 106 of Figure 1 can be used to implement a method for processing image data by dividing a frame of received image data into a plurality of non-overlapping subimages as illustrated in Figure 3a. As the frame of image data is divided, global information regarding the overall input frame of image data is retained and stored with each of the subimages as represented by the association of an image header 304 with each stored subimage in Figure 3b.

Global information can, for example, include information regarding the transformation matrix for transforming the input frame of image data into the output frame of image data, size of the image, image resolution and so forth. For example, the global information can identify the overall size of an image (e.g., 100 pixels by 100 pixels), and an origin reference location to which all other pixels in the image can be correlated. The origin reference location can, in an exemplary embodiment, be user specified to correspond with the pixel

location (0,0) in a two-dimensional space (e.g., x-y space) of the input frame buffer, representing the top left pixel in an image to which all subimages will be correlated. The global information of the input frame of image data can be stored with each divided subimage so that as the image is reconstructed in the output frame of image data, as shown by the reconstructed output frame 306 in Figure 3c, this information is available to enable precise reconstruction. Alternately, the global information can be stored in a memory location (e.g., look-up table) which is separate from the subimage but which is known in advance, or which is pointed to by a pointer of the subimage header. Those skilled in the art will appreciate that the global information can be stored in any location which can be identified and accessed at the time boundary information is to be used to partition the output frame buffer.

In addition, each subimage can be stored with local information which represents a relative position of the subimage with respect to the original frame of image data from which the subimage was derived. The frame of image data is divided according to a user-configured number of partitions. For example, the user can configure an input frame of image data corresponding to a 100 pixel by 100 pixel image into four hundred partitioned subimages of 5 by 5 pixels. The twenty subimages in the first row will have top left pixel locations at locations (0,0); (5,0); (10,0) ... (95,0), respectively, relative to the origin reference of (0,0). Each of the remaining 19 rows of subimages will be similarly correlated to the origin reference location. As with the global information, the local information can also be alternately stored in a memory location which is separate from the subimage, but which is known in advance, or which is pointed to by a pointer in the subimage header. Those skilled in the art will appreciate that the local information can be stored in any location which can be identified and accessed when boundary information is to be established for a given subimage.

The overall geometry information and the relative positional information can thus be used to collectively represent information which identifies boundaries between subimages. Each subimage is associated with header information used to identify common attributes regarding overall geometry of the original image, and which includes information that identifies a relative position of each subimage with respect to the overall geometry.

The relative position of each subimage with respect to the overall frame of input data, represented as a coordinate location of the top left pixel for each subimage in the above example, can be precomputed and stored at the time the original frame of input image data is subdivided into the plurality of

subimages. In addition to a reference coordinate location, the relative position identified in the header of each subimage can also include information that identifies boundaries between the subimage and all bordering subimages. This boundary information can, for example, include the local header information which identifies the relative position of each bordering subimage, and/or information which identifies the boundary pixel locations themselves. Alternately, the boundary information can be precomputed in the output frame buffer based on global information which identifies how the input image was partitioned. As mentioned previously, all or any portion, of such information can be stored directly in the header, or can be stored in the header as a pointer to some other location in memory.

Once the global and relative positional information has been calculated and stored as a header for all subimages in the input frame of image data, the frame of image data can be processed into subimage locations in memory. In accordance with exemplary embodiments, the entire frame of input image data can be stored and processed into subimage data, and then afterwards, mapped to the output frame buffer. Alternately, as subimage data is received, it can be processed to the output frame buffer without waiting for receipt of image data with respect to remaining subimages, and without concern that overall degradation in quality of an image associated with the output frame of image data will occur.

More particularly, as the image data associated with one or more subimages is being received by the input frame buffer, earlier stored image data can be mapped to the output frame buffer. Boundary information of the output frame buffer can be obtained from, or computed on the basis of, the subimage header information. Alternately, the boundary information can be incorporated into a bit map which is constructed from the header information and which identifies pixels in the input space that correspond to boundaries between subimages. For example, where boundaries in the output frame buffer are to be computed, an equation associated with the transformation matrix can be used to identify where the boundaries occur.

In the example above, where the transformation matrix specifies a scaling of the input image frame of data by an irrational number of ÷5, reference locations of various subimages relative to the origin reference location would include irrational pixel locations. That is, the reference position for each of the subimages in the first row of an output frame buffer would be at pixel locations of (0,0); (11.18,0); ... (212.42,0). Of course, no pixel locations can be mapped to irrational locations in the output frame buffer. It is

an inability of conventional forward mapping techniques to effectively deal with these boundary situations that results in holes in the output image. However, in accordance with exemplary embodiments, the boundaries for each subimage are mapped in advance to pixel locations which can be identified using the header information.

More particularly, when the first subimage of image data is being processed, the boundary between the first subimage and the second subimage in the output image is either acquired from memory or precomputed using information in the header. An irrational boundary is, in an exemplary embodiment, truncated to a rational number. Thus, for the first subimage, the boundary between the first and second subimages is (11.18,0), and would be truncated to a pixel location of 11. Similar boundary processing can be performed for other subimages adjacent the first subimage. For example, the boundary of the first subimage in the first row with respect to the first image in the second row, located at reference location (0,11.18), would also be truncated. Pixels of the first subimage in the input frame of image data can then be forward mapped to the pixel locations in the output frame of image data to create an 11 x 11 subimage (pixel positions 0Æ10 for both x and y directions) in the output frame buffer.

To process the second subimage of input data, the header information for the second subimage is retrieved from memory. Recall that the second subimage was referenced to the origin with the header (5,0) in the input frame of image data. Accordingly, when the value of 5 is scaled using the transformation matrix value of ÷5, it is immediately recognized that the irrational number of 11.18, representing the boundary between the first and second subimages, would have been truncated.

As a result, the second subimage is mapped from the pixel location 11 to the boundary between the second and third subimages. The boundary between the second and third subimages can be determined in the same manner used to determine the boundary between the first and second subimages. This boundary is represented by the pixel location which corresponds to 10 in the input frame of image data, scaled by the factor of ÷5, to a pixel location of (22.36,0) in the output frame. Because this is an irrational number, the second subimage can be mapped as a subimage having 11 columns of image data in the output frame (from location 11 to 21).

The foregoing process can be repeated, using relative positions of the subimages with respect to one another and with respect to the origin reference location to identify boundary locations in the output frame of data. Complete

In accordance with exemplary embodiments, wherein subimage information is stored with global information and relative placement information, enhanced processing of the input frame of image data can be performed on a subimage-by-subimage basis to an output device space without concern that quality of the reconstructed image in the output device space will be sacrificed. As a result, exemplary embodiments of the present invention can implement a transformation of a frame of input image data to a frame of output image data using a forward mapping from the input device space to the output device space without concern that the reconstructed image will include holes or boundary artifacts. The quality previously only associated with conventional inverse mapping techniques can therefore be achieved with a performance comparable to or better than that associated with conventional forward mapping techniques.

Those skilled in the art will appreciate that exemplary embodiments of the present invention can be embodied as software which can be used in conjunction with a processor, such as processor 106 of Figure 1. For example, exemplary embodiments of the present invention can be implemented as a computer readable medium which, in response to receipt of a frame of input image data, divides the image data into a plurality of non-overlapping subimages, and stores each divided subimage with information used to identify a relative position of the subimage with respect to an original image from which the frame of input image data was derived.

It will also be appreciated that scaling, as described above, can be implemented in any known fashion. For example, scaling can be implemented using pixel replication as described by Dale A. Schumacher in chapter 4 ("Fast Anamorphic Image Scaling") of the book entitled Graphic Gems II; Arvo, James; San Diego: Academic Press, Inc. 1991. It may be that a frame of image data has been generated at a resolution of 72 dots per inch. Using pixel replication, the image data can be scaled up to 150 dots per inch for subsequent transfer to the compression/decompression coprocessor. Alternately, the input image data for a given subimage can be expanded to a size greater than an expected size of the transformed image in the output frame buffer, and then cut back in the output frame buffer to conform with the boundaries of the subimage in the output frame buffer.

Those skilled in the art will appreciate that any scaling techniques can be used in accordance with exemplary embodiments of the present invention. Such scaling techniques can be used to either scale up or scale down an input frame of image data to any user selected resolution.

Further, those skilled in the art will appreciate that once partitioning of the input frame of image data has been performed and all headers created, any desired preprocessing of the frame of image data can be performed as part of the transformation. As mentioned previously, the preprocessing can include, for example, a resampling to match the input frame of image data to characteristics of an output device space.

Using exemplary input image processing techniques as described above, resampling of the frame of image data can be performed without affecting quality of an input image. Resampling can be included as part of the transformation matrix for producing the output frame of image data from the input frame of image data. When using resampling techniques, relative placement of subimages with respect to the overall frame of input data is retained in memory such that subimages in the output frame of image data can be properly placed and boundaries between subimages accurately located.

In accordance with exemplary embodiments, a frame of image data is resampled when the center of a predefined pixel block does not align with the center of a partitioned block. For example, it is possible that an incoming frame of image data is already at the desired resolution, such that scaling of the data is not necessary. However, the image data may not be aligned with the preselected partitions of the memory where the uncompressed is stored (for example, the frame buffer).

For example, the image data may be such that the first partitioned block of pixels in the frame buffer is only half full of data (that is, either the first two rows or the first two columns of a partitioned block contain no image data). In this case, the image data included within the pixel block is resampled such that the center of the resampled data is repositioned in alignment with the center of a partitioned block. This repositioning can be achieved by any conventional resampling technique. For example, resampling can be achieved using spatial filtering, nearest neighbor filtering, bi-linear interpolation, bi-cubic interpolation and so forth, without concern that the reconstructed image will have degraded quality.

Once preprocessing of the image data has been completed, those skilled in the art will appreciate that the partitioned output frame of image data can be subsequently processed in any way. For example, once partitioned, the output frame of image data can be compressed for storage and then later reconstructed from its compressed form. Where data is to be compressed, such compression and subsequent decompression can be performed in accordance with exemplary embodiments of the described in commonly assigned U.S.

Application Serial No. 08/397,372, filed March 2, 1995 (Attorney Docket No. P1481/149), entitled "METHOD AND APPARATUS FOR COMPRESSION OF DIGITIZED IMAGE DATA USING VARIABLE COLOR FIDELITY."

Exemplary embodiments of the present invention are directed to using a base resolution of 600 dots per inch, with each pixel including four color components of cyan, magenta, yellow and black. Each such color component, in exemplary embodiments, is one of 256 values. An entire pixmap, in accordance with exemplary embodiments is partitioned into non-overlapping blocks having a 4-by-4 pixel configuration. Thus, each partitioned block has a resolution of 150 dots per inch.

While the foregoing has set forth exemplary embodiments of the present invention, those skilled in the art will appreciate that these examples are by way of illustration only. For example, while exemplary embodiments have been described in connection with pixmaps used with an output printing device, those skilled in the art will appreciate that techniques of the present invention are also suitable for the reduction of information transmitted between computers, or between a computer and an input device such as a scanner, or between a computer and an output device (such as a printer).

Further, those skilled in the art will appreciate that the partitioned blocks described above can be of any dimension. Further, the partitioning can be adaptive and dynamic, even within a given frame of image data, with the only requirement being that dimensional information be made known to the decompression process.

It will be appreciated by those skilled in the art that the present invention can be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The presently disclosed embodiments are therefore considered in all respects to be illustrative and not restricted. The scope of the invention is indicated by the appended claims rather than the foregoing description and all changes that come within the meaning and range and equivalence thereof are intended to be embraced therein.

WHAT IS CLAIMED IS:

1. A method for processing image data, comprising the steps of:

receiving a frame of input image data;

dividing the image data into a plurality of non-overlapping subimages; and

storing each divided subimage with information used to identify a relative position of said divided subimage with respect to an image from which said frame of input image data was derived.

2. A method according to claim 1, wherein said step of storing further comprising the step of storing image characteristic information which identifies characteristics of the frame of input image data with each of said plurality of non-overlapping subimages.

3. A method according to claim 2, wherein said image characteristic information identifies a transformation matrix for transforming said frame of input image data into a frame of output image data.

4. A method according to claim 3, wherein said transformation matrix identifies a scaling factor of said frame of output image data relative to said frame of input image data.

5. A method according to claim 3 or 4, wherein said transformation matrix identifies a rotation of said frame of output image data relative to said frame of input image data.

6. A method according to one of claims 3 to 5, wherein said transformation matrix identifies a skew of said frame of output image data relative to said frame of input image data.

7. A method according to one of claims 2 to 6, wherein said image characteristic information identifies a size of said frame of input image data.

8. A method according to one of claims 2 to 7, wherein said image characteristic information identifies an image resolution of said frame of input image data.

9. A method according to one of claims 1 to 9, wherein said relative position information identifies boundaries between each of said plurality of non-overlapping subimages.

10. A method according to one of claims 1 to 9, wherein said frame of input image data includes a plurality of pixels, each of which is represented by a color value.

11. A method according to one of claims 1 to 10, wherein pixels of said frame of input image data are forward mapped to pixels in said frame of output image data.

12. An apparatus for processing image data comprising:

means for receiving a frame of input image data;

means for dividing the image data into a plurality of non-overlapping subimages; and

means for storing each divided subimage with information representing a relative position of said subimage with respect to an original image from which said frame of input image data was derived.

13. An apparatus according to claim 12, wherein said storing means further includes:

a frame buffer for storing at least a portion of said frame of input image data; and

a frame buffer for storing a frame of output image data produced by transforming said input image data.

14. A computer readable medium containing program instructions for:

dividing a frame of input image data into a plurality of non-overlapping subimages; and

controlling a storage of each divided subimage with information representing a relative position of said subimage with respect to an original image from which said frame of input image data was derived.

15. A method for processing image data, comprising the steps of:

receiving a frame of input image data;

dividing the frame of input image data into a plurality of non-overlapping subimages in an input space;

storing boundary information for each of said plurality of non-overlapping subimages in a memory associated with said input space; and

transforming said boundary information for each of said non-overlapping subimages into an output space.

16. A method according to claim 15 further including a step of:

transforming image data of at least one non-overlapping subimage from said input space to said output space after transforming said boundary information.

17. A method according to claim 15 or 16, wherein said step of storing further includes the steps of:

storing image characteristic information which identifies at least one characteristic of the frame of input image data; and

storing local information which identifies a relative position of each of said plurality of non-overlapping subimages with respect to a reference location of said frame of image data.

18. A method according to claim 17, wherein said image characteristic information identifies a transformation matrix for transforming said frame of input image data into a frame of output image data for said output space.

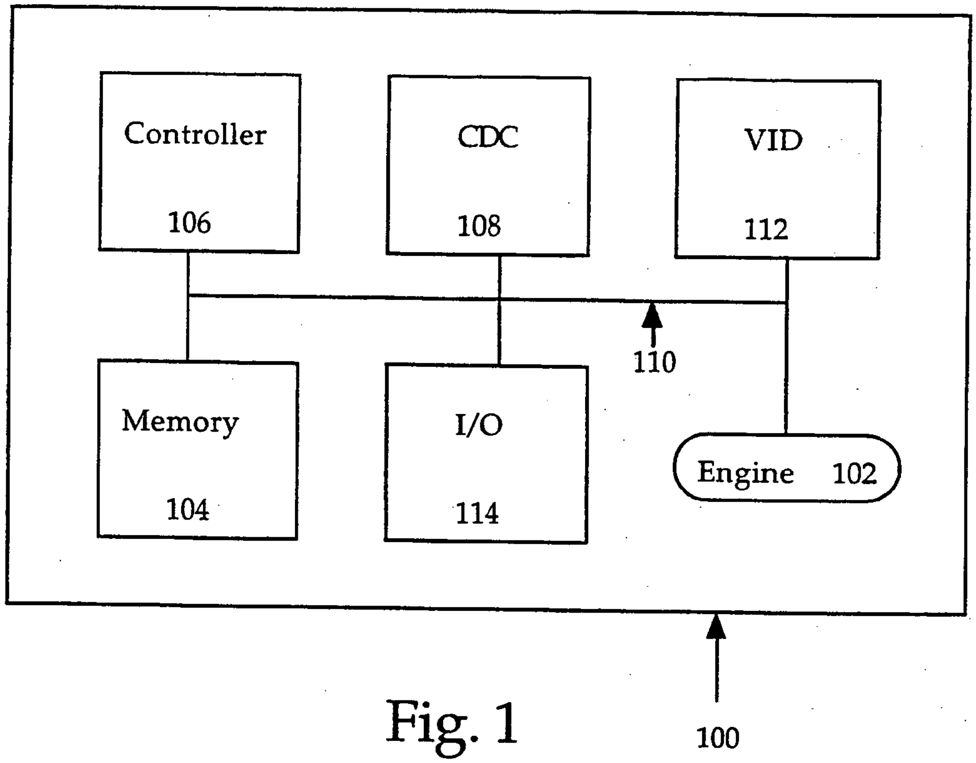


Fig. 1

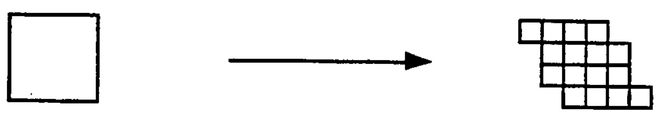


Fig. 2A

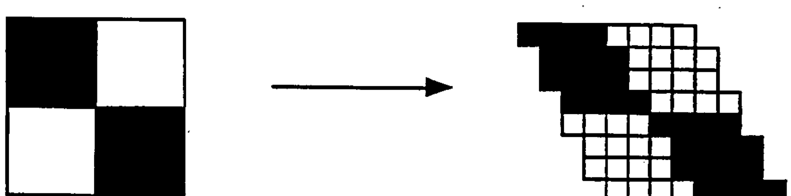


Fig. 2B

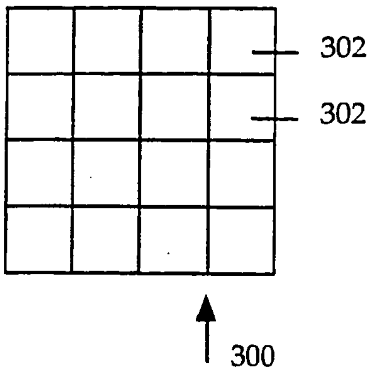


Fig. 3A

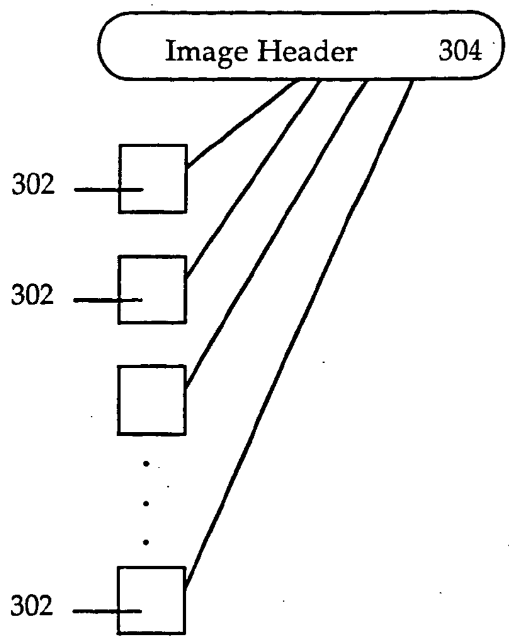


Fig. 3B

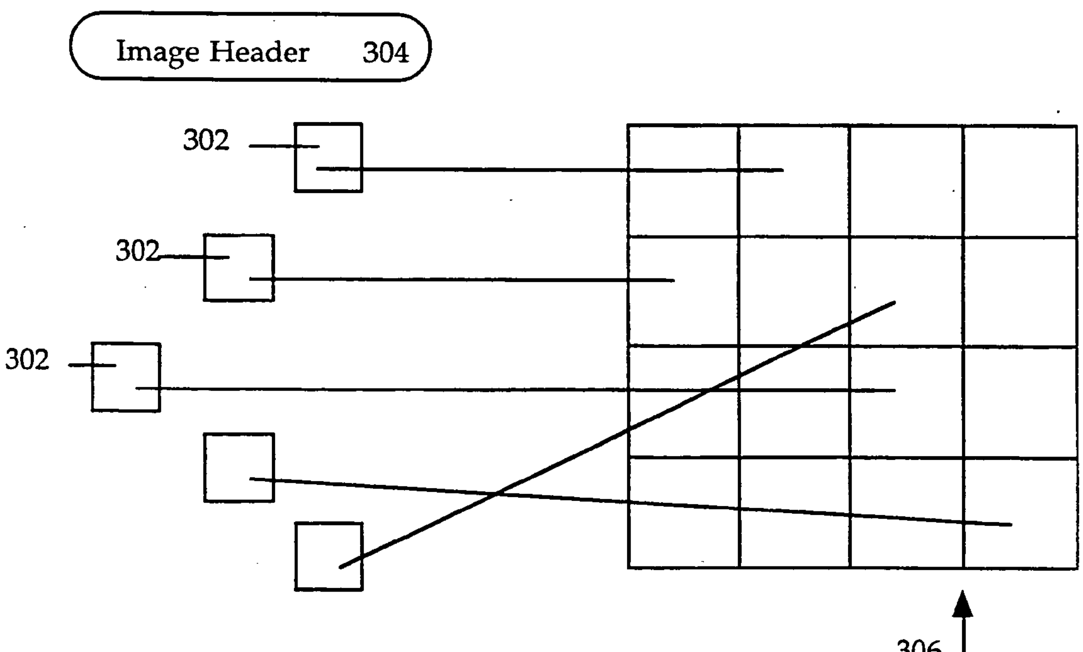


Fig. 3C

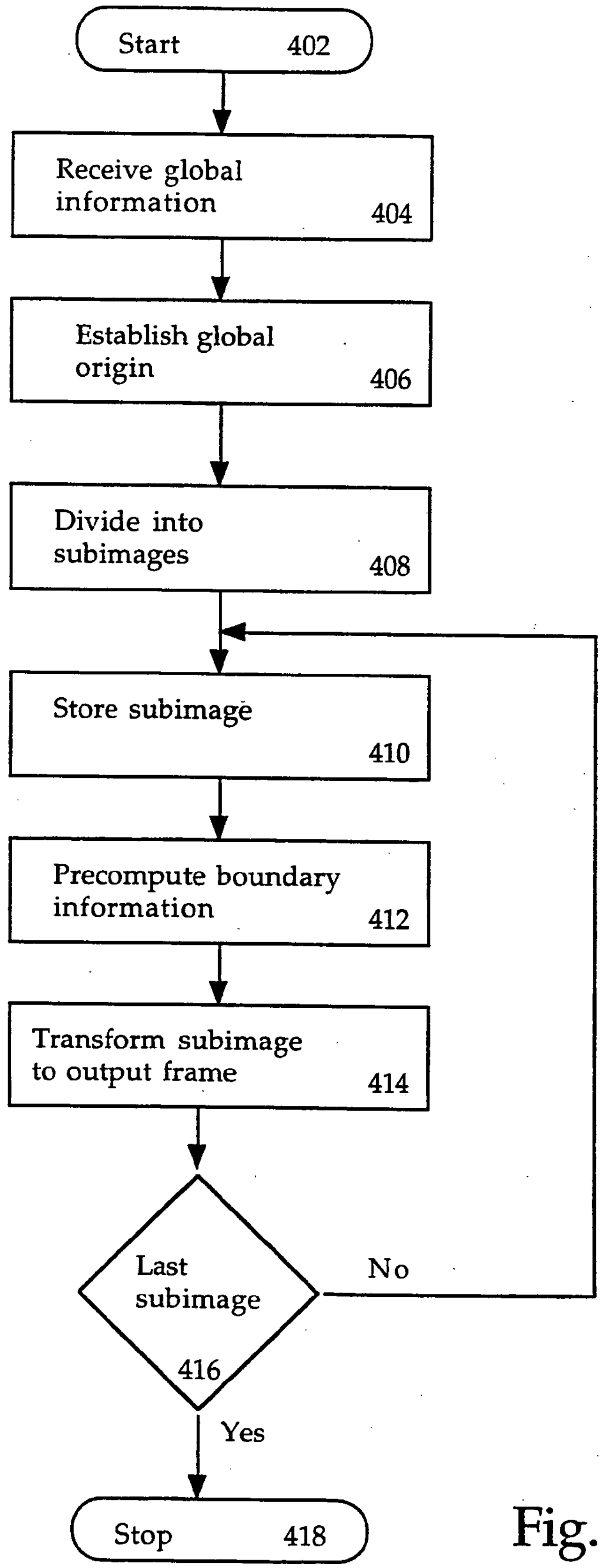

Start 402
Receive global information 404
Establish global origin 406
Divide into subimages 408
Store subimage 410
Precompute boundary information 412
Transform subimage to output frame 414
Last subimage 416
No
Yes
Stop 418

Fig. 4

INTERNATIONAL SEARCH REPORT

Inter nal Application No
PCT/US 96/08171

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06T9/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06T H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PROCEEDINGS OF THE FOURTH INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING APPLICATIONS AND TECHNOLOGY, PROCEEDINGS OF THE FOURTH INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING APPLICATIONS AND TECHNOLOGY. ICSPAT '93, SANTA CLARA, CA, USA, 28 SEPT.-1 OCT, 1993, NEWTON, MA, USA, DSP ASSOCIATES, USA, pages 658-661 vol.1, XP000576835 BAREKET R ET AL: "Fractal image compression using a Motorola DSP96002" see page 658, left-hand column, paragraph 2 see page 659, left-hand column, paragraph 5 --- -/--	1-18

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

2

Date of the actual completion of the international search	Date of mailing of the international search report
26 July 1996	01 August 1996

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Pierfederici, A

Form PCT/ISA/210 (second sheet) (July 1992)

Inter nal Application No
PCT/US 96/08171

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	PATENT ABSTRACTS OF JAPAN vol. 95, no. 009 & JP,A,07 250248 (TOSHIN KOGYO:KK), 26 September 1995, see abstract ---	1,12,14, 15
P,X	US,A,5 513 279 (YOSHINARI TOSHIAKI ET AL) 30 April 1996 see abstract; figure 6 see column 5, line 42 - line 67 ---	1,12,14, 15
X	WOLBERG G.: "Digital Image Warping" 1992 , IEEE COMP. SOC. PRESS , US, LOS ALAMITOS XP002009000 205940 cited in the application see page 190, paragraph 2; figure 7.2 see page 222, paragraph 2 ---	1-18
A	COMSIG 1989. PROCEEDINGS SOUTHERN AFRICAN CONFERENCE ON COMMUNICATIONS AND SIGNAL PROCESSING (CAT. NO.89TH0230-3), STELLENBOSCH, SOUTH AFRICA, 23 JUNE 1989, ISBN 0-87942-713-2, 1989, NEW YORK, NY, USA, IEEE, USA, pages 97-101, XP000576837 BARNARD G ET AL: "Reducing the block-effect in transform image coding" -----	

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter   al Application No
PCT/US 96/08171

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US-A-5513279	30-04-96	JP-A- 7079351	20-03-95

Form PCT/ISA/210 (patent family annex) (July 1992)

THIS PAGE BLANK (USPTO)